(1) Write a program to demonstrate Tightly Coupled code.

<DemoApplication.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

  public static void main(String[] args) {

    //Tight Coupling
    BookTicketImp bookTicketImp = new BookTicketImp();
    String source_dest = bookTicketImp.bookTicket("Delhi","Goa");
    System.out.println(source_dest);

    SpringApplication.run(DemoApplication.class, args);
  }

}
```

<BookTicketImp.java>

```
package com.ttn.spring.basics.demo;

public class BookTicketImp {

  //Tight Coupling
  public String bookTicket(String from,String to){
     AirTicket airTicket = new AirTicket();
     String ticket = airTicket.travelTicket(from,to);
     return ticket;
  }

}
```

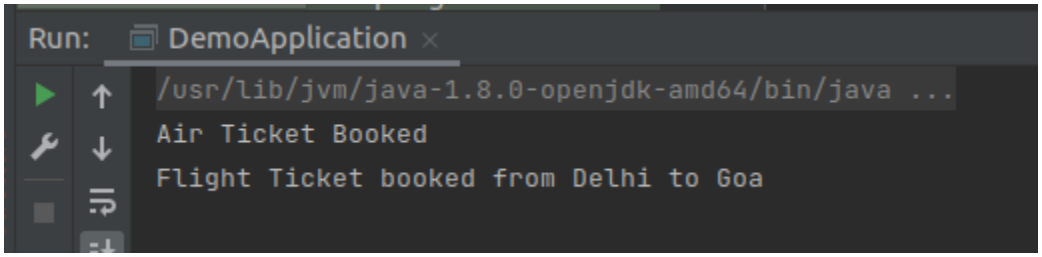
<AirTicket.java>

```
package com.ttn.spring.basics.demo;

public class AirTicket {
  public String travelTicket(String from,String to){
```

```
        System.out.println("Air Ticket Booked");
        return "Flight Ticket booked from " + from + " to " + to;
    }
}
```

<TrainTicket.java>

```
package com.ttn.spring.basics.demo;

public class TrainTicket {
    public String travelTicket(String from,String to){
        System.out.println("Train Ticket Booked");
        return "Train booked from " + from + " to " + to;
    }
}
```



(2) Write a program to demonstrate Loosely Coupled code.

<DemoApplication.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

  public static void main(String[] args) {

    //---Tight Coupling------
    // BookTicketImp bookTicketImp = new BookTicketImp();

    //Loose Coupling
    BookTicketImp bookTicketImp = new BookTicketImp(new AirTicket());
```

```
    String source_dest = bookTicketImp.bookTicket("Delhi","Goa");
    System.out.println(source_dest);

    SpringApplication.run(DemoApplication.class, args);
  }

}
```

<TravelMedium.java>

```
package com.ttn.spring.basics.demo;

public interface TravelMedium {
   public String travelTicket(String from,String to);
}
```

<BookTicketImp.java>

```
package com.ttn.spring.basics.demo;

public class BookTicketImp {
   TravelMedium travelMedium;

   BookTicketImp(TravelMedium travelMedium){
      this.travelMedium = travelMedium;
   }
   public String bookTicket(String from,String to){
      String ticket = travelMedium.travelTicket(from,to);
      return ticket;
   }

   /*
   //Tight Coupling
   public String bookTicket(String from,String to){
      AirTicket airTicket = new AirTicket();
      String ticket = airTicket.travelTicket(from,to);
      return ticket;
   }
    */

}
```

<AirTicket.java>

```
package com.ttn.spring.basics.demo;

public class AirTicket implements TravelMedium{
   public String travelTicket(String from,String to){
      System.out.println("Air Ticket Booked");
      return "Flight Ticket booked from " + from + " to " + to;
   }
}


/*
---------- Tight Coupling ------------------
public class AirTicket {
   public String travelTicket(String from,String to){
      System.out.println("Air Ticket Booked");
      return "Flight Ticket booked from " + from + " to " + to;
   }
} */
```

<TrainTicket.java>

```
package com.ttn.spring.basics.demo;

public class TrainTicket implements TravelMedium{
   public String travelTicket(String from,String to){
      System.out.println("Train Ticket Booked");
      return "Train booked from " + from + " to " + to;
   }
}


/*
------------ Tight Coupling ------------------
public class TrainTicket {
   public String travelTicket(String from,String to){
      System.out.println("Train Ticket Booked");
      return "Train booked from " + from + " to " + to;
   }
}
```

*/

(3) Use @Compenent and @Autowired annotations to in Loosely Coupled code for dependency management

<DemoApplication.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

@SpringBootApplication
public class DemoApplication {

  public static void main(String[] args) {

    //--- Tight Coupling ------
    // BookTicketImp bookTicketImp = new BookTicketImp();

    //------- Loose Coupling --------
    //BookTicketImp bookTicketImp = new BookTicketImp(new AirTicket());

    ApplicationContext applicationContext = SpringApplication.run(DemoApplication.class, args);
    BookTicketImp bookTicketImp = applicationContext.getBean(BookTicketImp.class);
    String source_dest = bookTicketImp.bookTicket("Delhi","Goa");
    System.out.println(source_dest);
  }

}
```

<BookTicketImp.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
```

```
@Component
public class BookTicketImp {
   TravelMedium travelMedium;

   @Autowired
   BookTicketImp(TravelMedium travelMedium){
      this.travelMedium = travelMedium;
   }
   public String bookTicket(String from,String to){
      String ticket = travelMedium.travelTicket(from,to);
      return ticket;
   }

   /*
   //Tight Coupling
   public String bookTicket(String from,String to){
      AirTicket airTicket = new AirTicket();
      String ticket = airTicket.travelTicket(from,to);
      return ticket;
   }
    */

}
```

<AirTicket.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.stereotype.Component;

@Component
public class AirTicket implements TravelMedium{
   public String travelTicket(String from,String to){
      System.out.println("Air Ticket Booked");
      return "Flight Ticket booked from " + from + " to " + to;
   }
}


/*
---------- Tight Coupling ------------------
public class AirTicket {
```

```
  public String travelTicket(String from,String to){
     System.out.println("Air Ticket Booked");
     return "Flight Ticket booked from " + from + " to " + to;
  }
} */
```

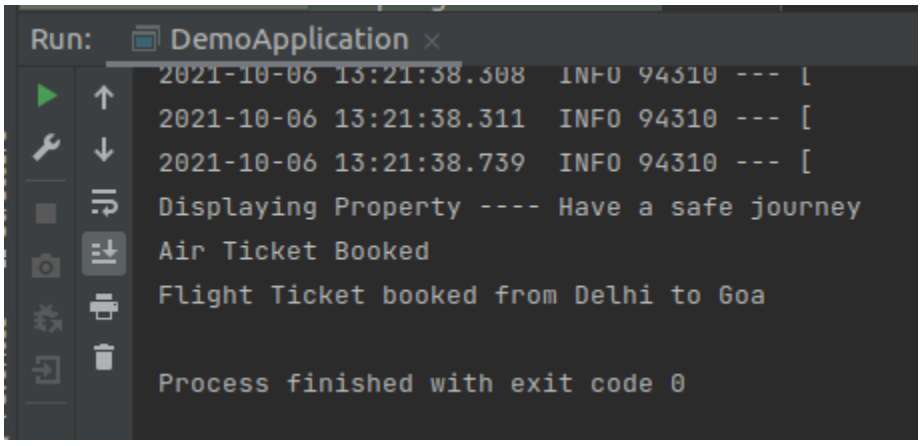
(4) Get a Spring Bean from application context and display its properties.

<BookTicketImp.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class BookTicketImp {
  TravelMedium travelMedium;

  @Value("Have a safe journey")
  String trip;

  @Autowired
  BookTicketImp(TravelMedium travelMedium){
     this.travelMedium = travelMedium;
  }
  public String bookTicket(String from,String to){
     String ticket = travelMedium.travelTicket(from,to);
     return ticket;
  }

  /*
  //Tight Coupling
  public String bookTicket(String from,String to){
     AirTicket airTicket = new AirTicket();
     String ticket = airTicket.travelTicket(from,to);
     return ticket;
  }
   */

}
```

<DemoApplication.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

@SpringBootApplication
public class DemoApplication {

  public static void main(String[] args) {

    //--- Tight Coupling ------
    // BookTicketImp bookTicketImp = new BookTicketImp();

    //------- Loose Coupling --------
    //BookTicketImp bookTicketImp = new BookTicketImp(new AirTicket());

    ApplicationContext applicationContext = SpringApplication.run(DemoApplication.class, args);
    BookTicketImp bookTicketImp = applicationContext.getBean(BookTicketImp.class);
    System.out.println("Displaying Property ---- " + bookTicketImp.trip);
    String source_dest = bookTicketImp.bookTicket("Delhi","Goa");
    System.out.println(source_dest);
  }

}
```



(5) Demonstrate how you will resolve ambiguity while autowiring bean (Hint : @Primary)

<AirTicket.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.context.annotation.Primary;
import org.springframework.stereotype.Component;

@Component
@Primary
public class AirTicket implements TravelMedium{
   public String travelTicket(String from,String to){
      System.out.println("Air Ticket Booked");
      return "Flight Ticket booked from " + from + " to " + to;
   }
}


/*
---------- Tight Coupling ------------------
public class AirTicket {
   public String travelTicket(String from,String to){
      System.out.println("Air Ticket Booked");
      return "Flight Ticket booked from " + from + " to " + to;
   }
} */
```

<TrainTicket.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.stereotype.Component;

@Component
public class TrainTicket implements TravelMedium{
   public String travelTicket(String from,String to){
      System.out.println("Train Ticket Booked");
      return "Train booked from " + from + " to " + to;
   }
}
```

```
/*
---------- Tight Coupling ----------
public class TrainTicket {
   public String travelTicket(String from,String to){
      System.out.println("Train Ticket Booked");
      return "Train booked from " + from + " to " + to;
   }
}

*/
```

(6) Perform Constructor Injection in a Spring Bean

<BookTicketImp.java>

```
package com.ttn.spring.basics.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class BookTicketImp {
   TravelMedium travelMedium;

   @Value("Have a safe journey")
   String trip;

   @Autowired
   //Constructor injection
   BookTicketImp(TravelMedium travelMedium){
      this.travelMedium = travelMedium;
   }
   public String bookTicket(String from,String to){
      String ticket = travelMedium.travelTicket(from,to);
      return ticket;
   }

   /*
   //Tight Coupling
```

```
    public String bookTicket(String from,String to){
        AirTicket airTicket = new AirTicket();
        String ticket = airTicket.travelTicket(from,to);
        return ticket;
    }
     */

}
```